Final Project: FroggySplat

Due Sunday, May 19

You may work with a partner on this project.

Using position, setTimeout, and parameters:

Note: The final project I came up with was to get the frog onto the lilypad while avoiding the alligators. You must use arrow keys to move the frog around on the board. If you get the frog onto the lilypad, you get 10 points. If you get eaten by an alligator, you turn into a splat and lose all your points. The biggest challenge in this game is figuring out when your frog is over your lilypad, and when it's over the alligator. You do that by comparing the x coordinates and y coordinates (from top left) for the lilypad, the frog, and also for the frog and the alligator. If both the x and y coordinates are within 15 pixels of each other, the frog is either on the lilypad, or the alligator is eating the frog.

If you want to make a cooler game, feel free. It must use positions, setTimeout, and parameters. Have fun. Make the next angry birds or fruit ninjas. Don't let me hold you back.

HTML Part:

To create the froggy game (or your version of it), create a web page with a lilypad image, a frog image, and an alligator image. Make sure all three images are the same width and height. Below the images, create a table with 4 arrows – a left arrow, an up arrow, a right arrow, and a down arrow. Each of these arrows will move the frog image left 10 pixels, up 10 pixels, down 10 pixels, or right 10 pixels, respectively. These 4 arrows will call your javascript function with a parameter that will hold 'left','up','down', or 'right' depending on which arrow you pic.

Also include a paragraph that will hold your score and a button to start your game.

JavaScript Part:

Now let's start on the JavaScript.

Inside the script (but outside the functions), create 7 variables (all initially set to 0):

- xposlily
- yposlily
- xposfrog
- yposfrog
- xposalli
- yposalli
- totalscore

(We want these outside the functions because we want more than one function to be able to use these variables.)

MoveFrog Function:

Now write a function that will move the frog to the left 10 pixels (you just need to subtract 10 from the xposfrog.) Make your left arrow image call this function. Make sure it works. Okay, now modify this function by adding a parameter. And modify your call to the function (in the left arrow) so that the call holds the word 'left'. And add a condition in the function saying, if the parameter holds the word 'left', then subtract 10 from xposfrog (this is a simple if condition.) Make sure this works.

Once you have this, you can modify it so that the right arrow image calls the same function, only with the word 'right'. Inside the function, add another condition: if the parameter holds the word 'right', add 10 to xposfrog. Make sure this works so that the right arrow moves the frog right when you click on it.

Do the same for the up and down arrows – if you click on the up arrow, call the function with 'up'. Then in the function if the parameter is 'up', subtract 10 from yposfrog. If you click on the down arrow, call the function with 'down'. Then in the function if the parameter is 'down', add 10 to the yposfrog.

Make sure all your arrows work to move the frog around. Once you have all your arrows working, you've got the crux of the program done.

MovePad function:

Now write a second function that will move the lilypad around the screen to random places. This is a short function, and a lot like functions you've already written for lab. Set xposlily to a random number between 0 and 800, and set yposlily to a different random number between 0 and 800. Set the left and top positions, respectively, to the xposlily and yposlily random variables. Then use setTimeout to recall the function every 20000 milliseconds (yes, that's a big number. I wanted the lilypad to stay still for a while. If you want to test this function, make it a smaller number. Then when you're sure it works, set it back to that long number.

Now go back and modify the *MoveFrog* function. You want to check to see if the frog is at the same position as the lilypad. You do that by looking to see if the xposfrog and yposlily and the yposfrog and the yposlily are within 15 pixels of each other. (e.g.,

if ((xpsfrog > (xposlily – 15))&& (xposfrog < (xposlily + 15)…

If they are within 15 points of each other you want to increase the totalscore by 10, and print it out to the paragraph that will hold the total score.

Alligator function:

Now you're going to write a function that moves an alligator across the screen. Alligators only swim in one direction, so this function will only change the xposalli. Since we're assuming our screen is about 800 pixels wide, you will only increase xposalli if xposalli < 650. If xposalli is less than 650, we want to increase the value by some small amount (depending on how slow and how smooth you want the alligator's movement to be). Then set the alligator's left position to be xposalli.

Now, just like you checked to see if the frog was over the lilypad, you want to check to see if the frog is at the same place as the alligator (in which case the alligator eats the frog and the frog loses all his points). Again, check to see if the frog is within 15 pixels of the alligator, and, if so, the totalscore gets set to 0.

And call setTimeout, with a relatively small time if the movement is small and a larger time if the movement is larger. You can play with these two values to get a very smooth-moving alligator, or a jerky-moving alligator. Your choice.

All this should only happen if the xposalli is < 650.

If xposalli >= 650, set the alligator image's src on the web page to be "", and set xposalli to 0 and yposalli to 0.

GetAlligator function:

The alligator function moves the alligator across the screen. This function creates an alligator every so many seconds (random), and figures out how far down in the screen the alligator should be (again, a random number), and then calls the alligator function. (Making the alligator appear at random times at a random distance from the top, and then move across the screen predatorily.)

More specifically, this function should generate a random number between 0 and 650 (assuming the screen is 800 pixels wide and the alligator image is 150 pixels high). Set xalli to be the random number. Change the alligator image on the web page's src to be the alligator.gif (or whatever you called it). Change the alligator's top position to be xalli. Now you have an alligator a certain distance from the top. Call the *alligator()* function (this will move the alligator across the screen).

Now set a basetime variable (you're just creating) to 30000. Set a time variable to be a random number between 0 and 10000 and add the basetime to it. Call setTimeout with this new time.

StartIt() function:

This is the final function, and it should consist of 2 lines: Calling the *MovePad* function and calling the *GetAlligator* function. This is the function that should be called when the start button in the web page is clicked on.

That's it! You've wrote the froggysplat game!

Extra Credit:

Extra Credit will be given for taking this game to the next level:

Simple upgrades (5 pts): making the frog's image change to a chomped image when the alligator eats the frog, making the frog's image change to a "ribbit" or something like that when he successfully get on the lilypad

Complex upgrade (20 pts): Make the game stop when the frog is eaten by the alligator, and a "game over" image appears.

If you can think of other cool upgrades to the game, let me know. They may be worth extra credit!